independent claim 26; and independent claim 27 and its dependent claim 28. In addition, the following claims stand rejected under 35 U.S.C. 103 (a) over Caron in view of U.S. Patent No. 5,819,283 to Turkowski ("Turkowski"): claims 4 and 7, which depend from claim 1; and claim 19, which depends from claim 16. Also, the following claims stand rejected under 35 U.S.C. 103 (a) over Turkowski: independent claim 11 and its dependent claims 12-15.

35 USC §103 (a) Rejection of the Claims over Caron

The Office action rejected claims 1, 3, 5-6, 10, 16-17, and 20-25 under 35 U.S.C. 103 (a) as being unpatentable over Caron. In particular, the Office action alleges that the reference "discloses a system that provides extended functionality from an extension to an extensible object . . . [by:] locating an extension, . . . obtaining an extension, . . . and directing references to the object that provides extended functionality in the extension." However, applicants respectfully disagree and traverse the rejection over Caron.

The distinction between Caron and the present invention is as fundamental as the distinction between an object and an instance of an object. As is well known by those skilled in the art, an object is a definition or a template for a type of data to be stored. An instance, on the other hand, is a tangible representation of the object that can be manipulated. Stated differently, an instance provides meaningful data (whose categories are defined by the object) that can be manipulated by a program. There can be more than

one instance of an object. Using analogy, for example, a "Person" object may have an instance called "Bob." The "Person" object may have defined categories like "height" and "weight," but it is the instance of the object that provides data for the categories. For example, the "Bob" instance may have "height = 60" and "weight = 150." This object/instance arrangement allows for multiple instances of an object to be created, for example, a "Jim" instance having "height = 70" and "weight = 200."

Caron is directed toward "instance customization." Instance customization refers to allowing users to customize the instances of an object that appear in an application program, for example. One way that Caron accomplishes such instance customization is by "put[ting] code behind application program documents to enable users to customize instances of objects that appear in an application program."

This is to be distinguished from the present invention. The present invention is directed to a method for extending functionality to an object in an object model. In other words, the present invention does not relate to the *instance* of the object, but rather to the *object itself*. As recited in claim 1, the present invention first locates a package having the desired extended functionality, and obtains an extension object related to the desired extended functionality. The present invention then directs certain references to the extended functionality.

For example, in one embodiment of the present invention, a hierarchical object model is extended. In particular, the methods of a window object 203 may be extended by an "editor" method via an editor extension object 204 (*Figure 2 and page 9, lines 15-*

23). The editor extension object 204 locates an editor package 206 that is registered on

an extension database 205, and obtains an extension object from the editor package.

When the window object 203 is first invoked by an application, it is determined that a

certain editor method is not inherent in the window object 203. The run-time

environment then searches the extension database 205 to locate the editor package 206.

The editor package 206 signals the extension provider object 208 to create the editor

extension object 204, so as to service all calls to the editor method.

Accordingly, because Caron is directed to customizing an instance of an object

and does not teach or suggest extending the functionality of an object (as with the present

invention), Applicants respectfully traverse the 35 U.S.C. 103 (a) rejection of claims 1, 3,

5-6, 10, 16, 17 and 20–25 over Caron. The argument above applies with equal force to

claims 2, 8, 9, 18, and 26-28 over Caron in view of Buxton; and to claims 4, 7, and 19

over Caron in view of Turkowski. Accordingly, Applicants respectfully traverse the

rejection of claims 2, 8, 9, 18, and 26-28 over Caron in view of Buxton, and of claims 4,

7, and 19 over Caron in view of Turkowski.

35 USC §103(a) Rejection of the Claims over Turkowski

Claims 11-15 stand rejected under 35 U.S.C. 103 (a) over Turkowski. In

particular, the Office action alleges that Turkowski discloses a metafile that contains

information of individual objects. The Office action asserts that the metafile includes

extensible object identifier data, an object field, an object name field, an object extension field, a version field, a description of an object, and a table for which an entry is referenced by a pointer. The Office action further notes that Turkowski does not expressly disclose an extension name field, an extension identifier field, and a friendly name field, but that it would have been obvious to one having ordinary skill in the art at the time the invention was made to apply names and identifiers to objects. Applicants, however, respectfully disagree and traverse the rejection over Turkowski.

Turkowski is to be distinguished from the present invention, because Turkowski is directed to adding data structures, while the present invention is directed to extending an object model. Turkowski describes objects in a metafile as collections of data that can be shared among various application programs (*Turkowski* – column 3, lines 41-43). The objects are self-identifying in that they contain information as to their size and function (*Id.* - lines 43-44). Essential data can be mapped to the objects as a sequence of data words that relate additional attributes to the objects after the object has been created (*Id.* – column 4, lines 45-51). Therefore, Turkowski is adding data structures.
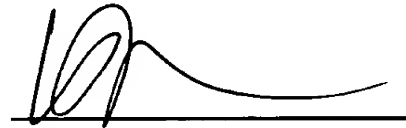
The present invention, on the other hand, is directed to extending an object model. In extending the object model, the present invention defines an extension identifier field, an extension name field, and an extension object identifier. Claim 11 has been amended to clarify this distinction, and has not been amended for the purposes of patentability. Accordingly, Applicants respectfully traverse the rejection of claims 11-15 over Turkowski.

CONCLUSION

In view of the foregoing amendments and remarks, Applicants respectfully submit that the present application is in condition for allowance. Reconsideration of the application and an early Notice of Allowance are respectfully requested. In the event that the Examiner cannot allow the present application for any reason, the Examiner is encouraged to contact Applicants' attorney at (215-564-8946).

Respectfully submitted,

Date: _2/3___, **2001**

Vincent J. Roccia

Registration No. **43,887**

WOODCOCK WASHBURN KURTZ
 MACKIEWICZ & NORRIS LLP
One Liberty Place - 46th Floor
Philadelphia, PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

Marked up versions of claim 11 amended herein, showing all of the changes relative to the previous version of each.

11. A computer-readable medium having stored thereon an entry for an object model extension [in extension data structure] comprising:

an extension identifier field containing data representing an identifier for the extension;

an extension name field containing data representing an external name for the extension by the extension identifier field; and

[a] an extension object identifier field containing data representing an extension package for the extension identified by the extension identifier field.